

Compte rendu TD4 - BDD - LECLERC

Sommaire

Introduction	1
Question 1)	2
Question 2)	2
Question 3)	3
Question 4)	3
Question 5)	3
Question 6)	4
Question 7)	4
Question 8)	4
Question 9)	4
Question 10)	5
Question 11)	5
Question 12)	5
Question 13)	6
Question 14)	6
Conclusion	6

Introduction

Ce TP a pour but d'appliquer les notions vues en cours sur les bases de données. Il permet de comprendre comment passer d'un MCD à un schéma relationnel, puis de créer et exploiter une base de données à l'aide du langage SQL.

Question 1)

Pour réaliser le Schéma Relationnel (MRD), j'ai analysé les cardinalités du MCD. Comme la cardinalité est de (1,1) côté Client et (0,N) côté Catégorie, j'ai fait migrer l'identifiant de la catégorie (numcat) comme clé étrangère dans la table CLIENT.

Le schéma relationnel est :

- CATÉGORIE (numcat, nomcat)
- CLIENT (num, nom, prenom, ville, #numcat)

Question 2)

Pour créer la structure de la base de données, j'ai utilisé l'instruction CREATE TABLE. J'ai défini les clés primaires (PRIMARY KEY) pour chaque table et la contrainte de clé étrangère (FOREIGN KEY) pour lier les clients aux catégories.

```
1 CREATE TABLE CATEGORIE (  
2     numcat INT PRIMARY KEY,  
3     nomcat VARCHAR(50)  
4 );  
5  
6 CREATE TABLE CLIENT (  
7     num INT PRIMARY KEY,  
8     nom VARCHAR(50),  
9     prenom VARCHAR(50),  
10    ville VARCHAR(50),  
11    numcat INT,  
12    FOREIGN KEY (numcat) REFERENCES CATEGORIE(numcat)  
13 );
```

Question 3)

Pour peupler la base de données, j'ai inséré les enregistrements fournis dans le sujet en utilisant la commande INSERT INTO pour les tables CATEGORIE puis CLIENT.

```

1 INSERT INTO CATEGORIE(numcat, nomcat)
2 VALUES(1, 'non renseigné'),(2, 'ouvrier'),(3, 'cadre'),(4, 'sans-emploi'),(5, 'autres');
3 INSERT INTO CLIENT(
4     num,
5     nom,
6     prenom,
7     ville,
8     numcat
9 )
10 VALUES(1, 'Qausi', 'Nathan', 'Paris', 1),(2, 'Vimif', 'Lucas', 'Paris', 3),(3, 'Gulya', 'Enzo', 'Courbevoie', 3),(4, 'Vetea', 'Léo', 'Nanterre', 4),(5, 'Lossu', 'Louis', 'Paris', 4),(6, 'Ebote', 'Emma',
    'Paris', 2),(7, 'Talam', 'Léa', 'Courbevoie', 3),(8, 'Julyn', 'Chloé', 'Paris', 2),(9, 'Voduk', 'Manon', 'Paris', 5),(10, 'Tuujji', 'Inès', 'Nanterre', 2);

```

Question 4)

Pour la question 4, j'ai créé une vue nommée Vue_NomCategories afin de n'afficher que le nom des catégories.

```

1 CREATE VIEW Vue_NomCategories AS
2 SELECT nomcat FROM CATEGORIE;

```

Question 5)

Pour la question 5, j'ai affiché la liste complète des clients avec une simple sélection de toutes les colonnes.

```

1 SELECT * FROM CLIENT;

```

Question 6)

Pour la question 6, j'ai créé une vue affichant les villes. J'ai utilisé le mot-clé DISTINCT pour éviter les doublons

```

1 CREATE VIEW Vue_VillesClients AS
2 SELECT DISTINCT ville FROM CLIENT;

```

Question 7)

Pour la question 7, j'ai créé une vue affichant les villes et les catégories présentes. J'ai fait une jointure entre les deux tables pour récupérer le nom de la catégorie.

```

1 CREATE VIEW Vue_VillesCategories AS
2 SELECT DISTINCT cl.ville, ca.nomcat
3 FROM CLIENT cl
4 JOIN CATEGORIE ca ON cl.numcat = ca.numcat;

```

Question 8)

Pour la question 8, j'ai déduit la ville ayant toutes les catégories (de 1 à 5). En analysant les données insérées, seule Paris possède des clients de tous les types de catégories.

```
1 SELECT ville FROM CLIENT
2 GROUP BY ville
3 HAVING COUNT(DISTINCT numcat) = (SELECT COUNT(*) FROM CATEGORIE);
```

Question 9)

Pour la question 9, j'ai créé une vue affichant le nom, le prénom et la catégorie des clients via une jointure.

```
1 CREATE VIEW Vue_ClientDetails AS
2 SELECT cl.nom, cl.prenom, ca.nomcat
3 FROM CLIENT cl
4 JOIN CATEGORIE ca ON cl.numcat = ca.numcat;
```

Question 10)

Pour la question 10, j'ai repris la requête précédente en ajoutant ORDER BY pour trier les résultats par nom.

```
1 SELECT cl.nom, cl.prenom, ca.nomcat
2 FROM CLIENT cl
3 JOIN CATEGORIE ca ON cl.numcat = ca.numcat
4 ORDER BY cl.nom ASC;
```

Question 11)

Pour la question 11, j'ai filtré la table pour n'afficher que les clients habitant à "Courbevoie".

```
1 SELECT nom, prenom
2 FROM CLIENT
3 WHERE ville = 'Courbevoie';
```

Question 12)

Pour la question 12, j'ai affiché les clients de la catégorie "Cadre" en faisant une jointure pour filtrer sur le nom de la catégorie.

```
1 SELECT cl.nom, cl.prenom
2 FROM CLIENT cl
3 JOIN CATEGORIE ca ON cl.numcat = ca.numcat
4 WHERE ca.nomcat = 'cadre';|
```

Question 13)

Pour la question 13, j'ai utilisé la fonction d'agrégation COUNT(*) pour compter le nombre de clients à Courbevoie.

```
1 SELECT COUNT(*) FROM CLIENT WHERE ville = 'Courbevoie';|
```

Question 14)

Pour la question 14, j'ai compté le nombre de cadres en utilisant une jointure et la fonction COUNT(*).

```
1 SELECT COUNT(*)
2 FROM CLIENT cl
3 JOIN CATEGORIE ca ON cl.numcat = ca.numcat
4 WHERE ca.nomcat = 'cadre';|
```

Conclusion

Ce TP m'a permis de mieux comprendre la création et l'utilisation d'une base de données relationnelle. Les différentes requêtes réalisées m'ont aidé à consolider mes bases en SQL et à mieux maîtriser les jointures, les vues et les fonctions de comptage.